Железкой по сети. Пробрасываем USB-устройства, видеокарту, звуковуху и диски на удаленную машину

Илья Остапенко , 29.12.2017 1 комментарий 140,938 Добавить в закладки



Содержание статьи

01. USB
02. RS232
03. Видеокарта
04. Звуковая карта
04.1 Альтернативный вариант
05. Диски
06. Выводы

В одной из прошлых статей мы обсуждали способы подключения самых разных сущностей как файлов и каталогов: WebDAV, BitTorrent, SSH и даже память видеоадаптера. Но что, если мы хотим получить доступ не к удаленному или локальному сервису, а к устройствам удаленной машины? Скажем, пробросить на локальную машину USB-порт и использовать подключенные к нему устройства как локальные.

Особенность Unix-подобной системы — относиться к любому из своих компонентов как к файлу — давно уже стала общим местом в разговорах о ее внутреннем устройстве. И бесчисленное количество статей о том же Linux тому свидетельство. Оборудование — не исключение. Видеокарта, аудиокарта, внешний девайс, подключенный через USB, в понимании Linux не что иное, как файл.

Оттого удивительно, что из всех операционных систем только Plan 9 (если не считать пары отпочковавшихся проектов со схожей судьбой), в котором подобный подход доведен до логического конца, способен без лишних телодвижений распознавать оборудование удаленного компьютера и управлять им, как своим собственным.

В Plan 9 за проброс оборудования отвечает RPC-протокол 9P. Он обеспечивает доступ вообще к любым файлам и устройствам, как локальным, так и сетевым. К сожалению, Linux похвастать таким универсальным инструментом не может. Зато здесь есть несколько инструментов (если не сказать — костылей), обеспечивающих доступ к оборудованию удаленной машины.

USB

Когда речь заходит о пробросе оборудования на другой компьютер, возможно, первое, что приходит на ум, — это веб-камера на

домашнем ноутбуке или подключенный к нему смартфон, доступ к которым нужно обеспечить с удаленного десктопа. Например, из офиса на другом конце города (в другом городе, в другой стране).

В подобном случае выручить может утилита USB/IP. Развитием утилиты уже давненько никто не занимался, но на ее работоспособности это пока не сказалось — в репозиториях большинства популярных дистрибутивов такой пакет присутствует.

Первым делом пакет USB/IP следует установить на ту машину, доступ к устройствам которой необходимо получить извне. Далее загружаем необходимые модули:

```
$ sudo modprobe usbip-core
$ sudo modprobe usbip-host
```

Проверяем, все ли корректно загрузилось:

```
$ sudo lsmod | grep usbip
```

И запускаем сервер:

```
$ sudo usbipd -D
```

Поскольку USB/IP имеет собственную, независимую от встроенной систему адресации, поиск устройств выполняется командой

```
$ sudo usbip list -l
```

Она покажет список всех устройств, подключенных в данный момент в USB-шине.

```
jacobtey@jacobtey-laptop ~ % sudo modprobe usbip-core
* pam_usb v0.5.0
* Authentication request for user "jacobtey" (sudo)
* Device "FLASHKEY" is connected (good).
* Performing one time pad verification...
* Access granted.
jacobtey@jacobtey-laptop ~ % sudo modprobe usbip-host
jacobtey@jacobtey-laptop ~ % sudo lsmod | grep usbip
usbip_host             24576  0
usbip_core             28672  1 usbip_host
usbcore               200704  11 uas,uhci_hcd,snd_usb_audio,uvcvideo,usb_storage,snd_usbmidi_lib,ehci_hcd,ehci_pci,usbhid,usbip_host,cdc_acm
usb_common             16384  2 usbip_core,usbcore
jacobtey@jacobtey-laptop ~ % sudo usbipd -D
jacobtey@jacobtey-laptop ~ % sudo usbip list -l
 - busid 5-1 (0d8c:013c)
   C-Media Electronics, Inc. : CM108 Audio Controller (0d8c:013c)

 - busid 7-2 (276d:1061)
   unknown vendor : unknown product (276d:1061)

 - busid 2-2 (04e8:6860)
   Samsung Electronics Co., Ltd : Galaxy (MTP) (04e8:6860)

 - busid 2-3 (064e:a101)
   Suyin Corp. : Acer CrystalEye Webcam (064e:a101)

 - busid 2-4 (1005:b113)
   Apacer Technology, Inc. : Handy Steno/AH123 / Handy Steno 2.0/HT203 (1005:b113)

jacobtey@jacobtey-laptop ~ % _
```

jacobtey : zsh

Настройка утилиты USB/IP

Теперь можно приступить непосредственно к расшариванию девайса (допустим, это будет веб-камера с индексом 2-3 из полученного списка):

```
$ sudo usbip bind --busid=2-3
```

Очередная проверка правильности выполненных действий:

```
$ sudo usbip list -r localhost
```

С исходным компьютером покончено. Далее следует настроить тот, на котором будет использоваться периферия первого.

Итак, перейдя на клиентскую машину, устанавливаем на нее USB/IP и запускаем:

```
$ sudo modprobe usbip-core
$ sudo modprobe vhci-hcd
```

Проверяем доступность расшаренного оборудования на сервере по списку:

```
$ sudo usbip --list АДРЕС_СЕРВЕРА
```

И присоединяем нашу камеру:

```
$ sudo usbip --attach АДРЕС_СЕРВЕРА 2-3
```

Проверяем результат:

```
$ sudo usbip --port
```

Теперь удаленное USB-устройство должно появиться в списке локальных, и с ним можно будет работать, как с любым другим. Для проверки корректности подключения выполняем команду lsusb:

```
$ lsusb
```

INFO

Существует клиентская часть USB/IP для Windows. Однако из-за ошибки в версии протокола в ее исполняемом файле из коробки она работает некорректно. Требуются дополнительные телодвижения с подменой констант в исходниках.

RS232

Самым лаконичным решением взаимного расшаривания в Линуксе могут похвастаться COM-порты. Никакие дополнительные драйверы для этого не нужны. За все отвечает одна маленькая утилита remserial, доступная в исходниках. Подходит как для доступа из Линукса к оборудованию, подключенному через RS232 на удаленном компьютере, так и для связки двух девайсов с COM-портами, подключенных к разным машинам, связанным по сети.

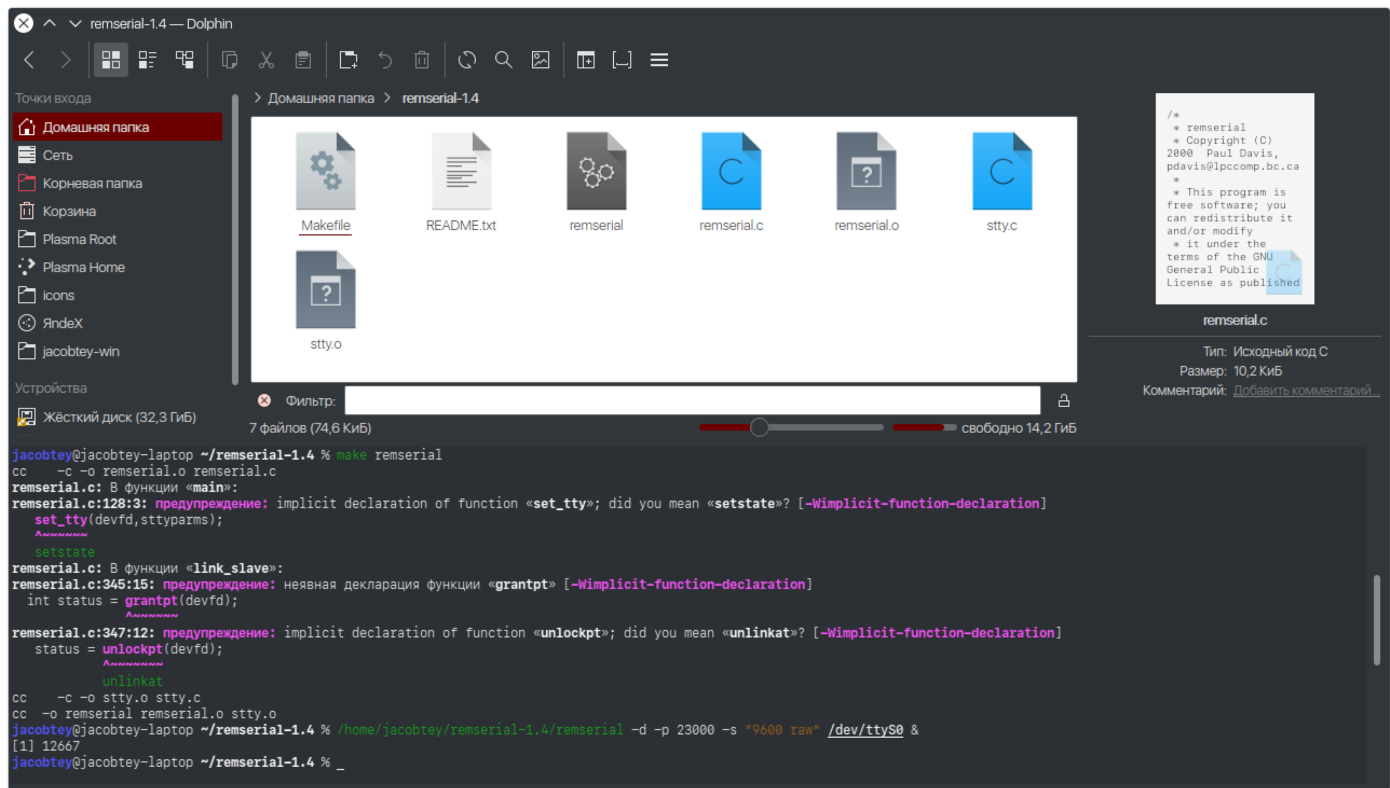
Расшарить RS232, указав сетевой порт (-p), скорость, режим stty (-s) и имя порта (здесь /dev/ttyS0), можно так:

```
$ remserial -d -p 23000 -s "9600 raw" /dev/ttyS0 &
```

Подключиться к COM-девайсу, расположенному на удаленной машине (сервере), — так:

```
$ remserial -d -r адрес_сервера -p 23000 -s "9600 raw" /dev/ttyS0 &
```

Допустимо запускать несколько экземпляров программы с разными портами и адресами подключенных девайсов.



Компиляция и запуск remserial

Видеокарта

Собираясь расшаривать видеокарту, уместно задать себе вопрос: зачем это нужно? Ведь недостатка готовых решений от проприетарщиков нет. Тут тебе и TeamViewer, совсем недавно порадовавший линуксовым клиентом, и облачные решения, встраиваемые в браузер. Однако не всегда все так гладко.

Видеоподсистемы современных ноутбуков повсеместно стали делать гибридными, когда на борту присутствуют две видеокарты: первая — мощная, но безголовая — дискретная, которая используется для трудоемких задач и включается по мере надобности для трансляции потоков через вторую — скромную интегрированную, работающую постоянно.

Масштабировав ситуацию до сетевых размеров и представив себе, что мощная видеокарта находится на удаленном компьютере, а в непосредственном доступе пользователя лишь старенький нетбук, для которого уже и запуск современного браузера порой непозволительная роскошь, приходишь к решению, взятому на вооружение все в той же гибридной графике. Да еще и инструментарий тот же — VirtualGL.

INFO

На самом деле все наоборот, и VirtualGL изначально задумывался для сетевых решений, а хинт с гибридами созрел потом.

Пакет VirtualGL присутствует в большинстве популярных дистрибутивов, поэтому все, что нужно сделать, — это установить его штатным менеджером пакетов на обеих машинах и, в зависимости от того, в какой каталог установлен пакет, запустить на сервере конфигуратор. В Arch Linux и его производных это делается командой

```
$ sudo vglserver_config -config +s +f -t
```

Если в твоей системе пакет установлен в `/opt`, нужно прописать полный путь:

```
$ /opt/VirtualGL/bin/vglserver_config -config +s +f -t
```

Далее перезапускай дисплейный менеджер, и сервер можно оставлять в покое.

```
jacobtey@jacobtey-laptop ~ % sudo usbip bind --busid=2-3
usbip: info: bind device on busid 2-3: complete
jacobtey@jacobtey-laptop ~ % sudo usbip list -r localhost
Exportable USB devices
======================
 - localhost
        2-3: Suyin Corp. : Acer CrystalEye Webcam (064e:a101)
           : /sys/devices/pci0000:00/0000:00:1d.7/usb2/2-3
           : Miscellaneous Device / ? / Interface Association (ef/02/01)

jacobtey@jacobtey-laptop ~ % _
```

jacobtey : zsh

jacobtey : zsh

Работа конфигуратора VirtualGL

На клиентской машине после установки VirtualGL в нем самом ничего конфигурировать не надо. Здесь нам понадобятся только vglconnect и vglclient. Зато нужно установить и настроить SSH. Открываем файл настроек `/etc/ssh/sshd_config` и добавляем в него такую строку:

```
X11Forwarding yes
```

Перезапускаем сервис sshd. В системах, основанных на systemd, это можно сделать так:

```
$ systemctl start sshd
```

Получить изображение запущенной на сервере программы и тем самым проверить работоспособность связки позволит команда

```
$ ssh -X логин@сервер НАЗВАНИЕ_ПРОГРАММЫ
```

Теперь можно с помощью vglconnect подключиться к серверу, попутно зашифровав трафик X11 и изображение VGL:

```
$ vglconnect -s логин@сервер
```

и попробовать запустить какое-нибудь приложение:

```
$ vglrun -q 40 -fps 25 НАЗВАНИЕ_ПРОГРАММЫ
```

Значения `-q` (степень сжатия JPEG) и `-fps` (частоту обновления кадров) можно варьировать, подбирая максимально возможные на используемом канале (не резиновом). Сжатие через JPEG при удаленном доступе используется по умолчанию.

При желании вернуть систему в исходное состояние поможет команда

```
$ sudo vglserver_config -unconfig
```

Звуковая карта

До появления звука в кино оно оставалось немым без малого тридцать лет. У нас времени чуть меньше, поэтому вслед за видео приступим к пробросске аудио. Здесь все еще проще, если воспользоваться возможностями вездесущего PulseAudio.

На клиенте и сервере открываем конфигурационный файл `~/.config/pulse/default.pa` или `/etc/pulse/default.pa` и раскомментируем строку `load-module module-native-protocol-tcp`. На сервере также добавляем одну из следующих строк:

```
auth-ip-acl=127.0.0.1 # Для доступа клиента с конкретным IP
auth-anonymous=true   # Для доступа с любого IP
```

Также на клиенте открываем конфиг `~/.config/pulse/client.conf` или `/etc/pulse/client.conf` (в зависимости от того, нужно ли установить эти настройки для одного пользователя или для всех) и добавляем:

```
default-server = адрес_сервера
```

Вместо редактирования `client.conf` для вывода на удаленный сервер можно использовать утилиту pax11publish:

```
$ pax11publish -e -S адрес_сервера
```

Отключение вывода на удаленную машину:

```
$ pax11publish -e -r
```

Чтобы изменения вступили в силу для конкретной программы, использующей PulseAudio, требуется ее перезапустить.

Альтернативный вариант

В дебрях интернета в слегка запущенном виде можно найти проект аудиотрансляции по сети — TRX, основанный на ALSA, кодеке Opus и библиотеке oRTP. Программа проста в использовании, как автомат Калашникова, но уступает ему в безотказности — исходники не обновлялись с 2014 года. Есть подозрения, что это касается лишь публично доступных материалов, поскольку основанный на нем онлайн-сервис аудиообмена от того же автора Cleanfeed выглядит достаточно свежо.

Диски

Для многих термин «расшаривание» прочно ассоциируется именно с открытием доступа к дисковым накопителям, их разделам и хранящейся на них информации. Даже неискушенные пользователи могут похвастаться знанием слова Samba (и не боятся вбить это слово в поисковик, чтобы найти краткий туториал по быстрой настройке, попутно натыкаясь на новости об очередной обнаруженной уязвимости и массовых эпидемиях, вызванных коварной вирусней).

Тому, кто привык находить нетривиальные решения, наверняка будет интересен вариант экспорта дискового устройства, по способу реализации приближенный к промышленным стандартам. В рамках возможностей Linux подобный экспорт делается при помощи программной системы DRBD, с 2009 года вошедшей в само ядро. Суть системы отражена в ее названии — распределенное реплицируемое блочное устройство. Она обеспечивает мгновенную синхронизацию между локальным и удаленным блочными устройствами.

Подготовительная работа заключается в установке пакета drbd-utils. Конфигурировать его можно пошагово с помощью drbdsetup, но проще отредактировать содержимое файла настроек `/etc/drbd.conf` **на обеих машинах:**

```
global { usage-count no; }
common { resync-rate 100M; }
resource r0 {
  protocol C;
  startup {
    wfc-timeout  15;
    degr-wfc-timeout 60;
  }
  net {
    cram-hmac-alg sha1;
    shared-secret "ПАРОЛЬ";
  }
  on имя_хоста1 {
    device /dev/drbd0;
    disk /dev/sda5;
    address 192.168.0.1:7788;
    meta-disk internal;
  }
  on имя_хоста2 {
    device /dev/drbd0;
    disk /dev/sda7;
    address 192.168.0.2:7788;
    meta-disk internal;
  }
}
```

Опцией cram-hmac-alg задается алгоритм шифрования, поддерживаемый на обоих компьютерах. Они перечислены в файле `/proc/crypto`**. Пароль shared-secret может содержать до 64 знаков.**

Имена хостов определяются командой `uname -n`**, запускаемой на каждой из машин отдельно. Опциям disk присваиваются названия дисковых накопителей или их разделов, относящихся к каждой машине. Следом указываются IP-адреса и задействованные порты. Их размер пропорционален размеру самого устройства.**

Также на обеих машинах запускаем инициализацию хранилища:

```
$ sudo drbdadm create-md drbd0
```

И стартуем демон DRBD:

```
$ sudo systemctl enable drbd
$ sudo systemctl start drbd
```

Далее переходим на ту машину, которая будет работать с данными (одновременный доступ к хранилищу может получить только одна из них), и делаем ее мастером:

```
$ sudo drbdadm primary all
```

Затем создаем и монтируем файловую систему хранилища:

```
$ sudo mkfs.ext4 /dev/drbd0
$ sudo mount /dev/drbd0 /mnt
```

В итоге получился своего рода RAID-массив поверх сети с единовременным доступом к нему только одного из существующих хостов. Впрочем, получить доступ к данным, находясь на другой машине, также возможно. Для этого нужно всего лишь размонтировать файловую систему на текущем мастере, сделать мастером вторую машину и смонтировать хранилище на ней. Трех элементарных команд, одну из которых можно повесить на сочетание клавиш сразу, а две другие предварительно вставить в сценарий, будет достаточно.

Выводы

Линукс не идеален и временами может серьезно проигрывать другим системам по своим функциональным возможностям, в точности как в случае с системой Plan 9 и удаленным доступом к аппаратным ресурсам компьютера. В то же время, когда возникает потребность решить конкретную задачу, шансы найти подходящий набор инструментов для этого практически стопроцентные.

Оцени статью:





Илья Остапенко

Теги: Linux USB Выбор редактора Статьи